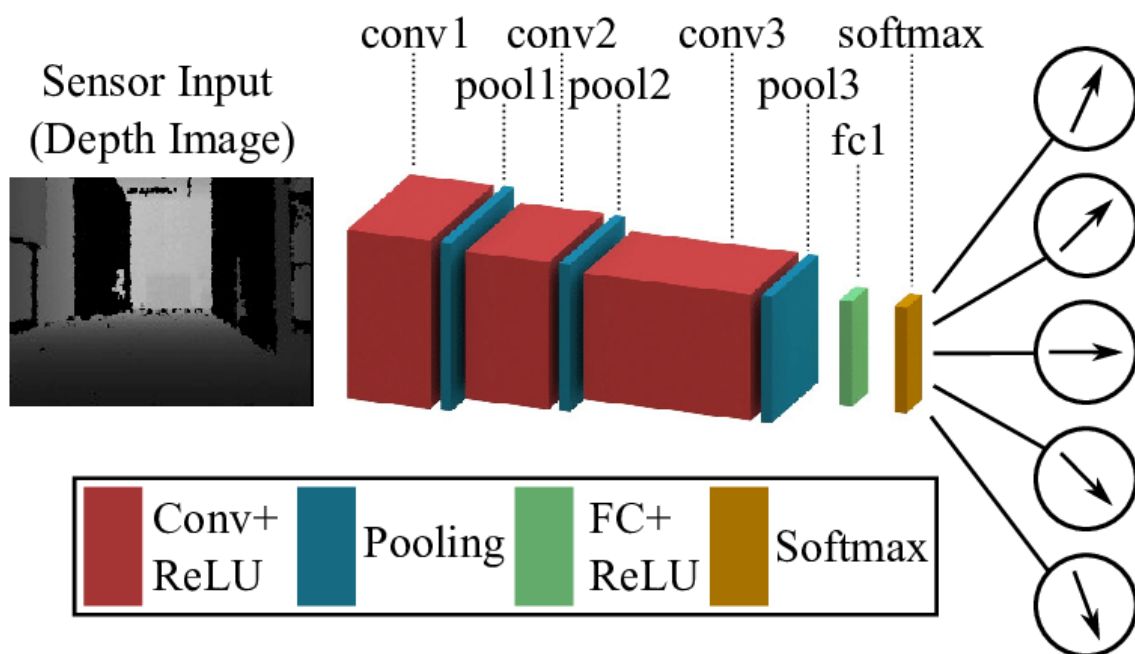


DEEP LEARNING EN ROBÓTICA MÓVIL



Pablo Calvo Arpón

Darío Suárez Jiménez

Javier Soler Villora

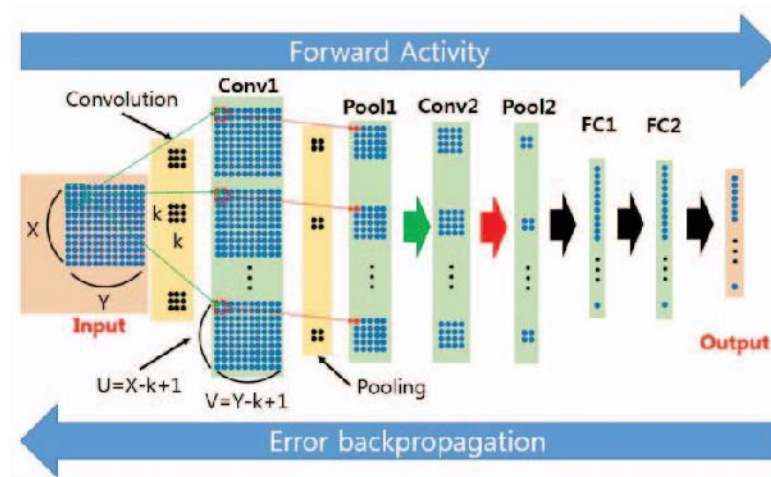
1. Deep Learning

El deep learning es una técnica de aprendizaje automático que se basa en el uso de redes neuronales profundas para modelar y resolver problemas complejos. La robótica móvil es un campo en el que el deep learning ha tenido un gran impacto, ya que los robots móviles necesitan ser capaces de percibir y navegar en su entorno de manera efectiva para llevar a cabo tareas útiles.

2. Las CNN

Uno de los usos más comunes del deep learning en la robótica móvil es el uso de redes neuronales convolucionales (CNN) para el procesamiento de imágenes. Las CNN son un tipo específico de red neuronal que se utiliza para detectar patrones y características en imágenes. Estas redes están compuestas por varias capas, cada una de las cuales se especializa en la detección de patrones específicos en la imagen de entrada. A medida que la imagen se desplaza a través de las capas de la red, se extraen características cada vez más complejas, como formas y texturas. Esto permite que la red sea capaz de detectar objetos y características en una imagen de manera robusta, independientemente de su posición, orientación o iluminación.

Para entrenar una CNN, se necesita un gran conjunto de datos de entrenamiento etiquetado, como imágenes con objetos específicos marcados y su posición. Los data sets que se utilizan para entrenar estas redes pueden variar según el objetivo de la tarea, pero algunos de los más populares incluyen COCO (Common Objects in Context) (Lin et al., 2014), PASCAL VOC (Visual Object Classes) (Everingham et al., 2010) y ADE20K (Zhou et al., 2017). Estos conjuntos de datos son alimentados a través de la red neuronal, y los pesos de las conexiones entre las neuronas son ajustados para minimizar el error en las predicciones mediante el uso de algoritmos de optimización como Adam o SGD. Estos algoritmos ajustan los pesos de las conexiones entre las neuronas a medida que la red neuronal se va entrenando.



3. Aplicaciones a la robótica móvil

En robótica móvil, las CNN se utilizan principalmente para tareas de percepción, como el reconocimiento de objetos y la detección de obstáculos. Esto permite que los robots móviles puedan percibir y entender su entorno, lo que es esencial para llevar a cabo tareas útiles como la navegación y el manejo de objetos. En el reconocimiento de objetos, se utilizan CNN para clasificar objetos en diferentes categorías, como personas, vehículos y animales. En la detección de obstáculos, se utilizan CNN para detectar objetos o características en una imagen que indican la presencia de un obstáculo.

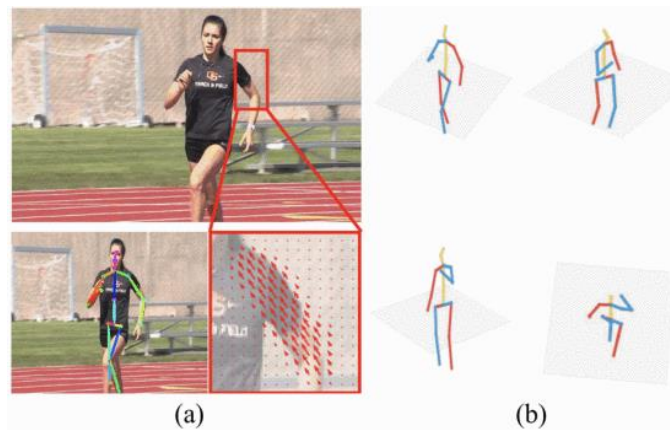
Además de las tareas de percepción, las CNN también se utilizan para la localización y navegación de robots móviles mediante el uso de algoritmos de visión por ordenador para el reconocimiento de características del entorno, tales como marcas y puntos de referencia. Esto permite a los robots móviles navegar de manera autónoma en un entorno conocido, siguiendo marcas en el suelo o reconociendo puntos de referencia.

Sin embargo, el deep learning en robótica móvil todavía tiene sus desafíos, como la capacidad de las CNN para operar en condiciones de iluminación y climáticas adversas, la adaptación a diferentes entornos, la capacidad para realizar tareas en tiempo real y la posibilidad de realizar tareas específicas en escenarios no previstos. Además, la interpretabilidad de las decisiones tomadas por una red neuronal profunda puede ser un problema, ya que las redes neuronales complejas son difíciles de entender y explicar.

4. Estado del arte

4.1. Pose Estimation using Part Affinity Fields

El paper "Real-time Multi-person 2D Pose Estimation using Part Affinity Fields" presenta una técnica para el reconocimiento de poses de personas en tiempo real mediante el uso de una red neuronal convolucional (CNN) y una técnica denominada campos de afinidad de partes. La idea principal detrás de la técnica es utilizar una CNN para detectar y localizar diferentes partes del cuerpo humano en una imagen, como cabeza, cuello, hombros, caderas, rodillas, etc. y luego utilizar esta información para estimar la posición completa de cada persona en la imagen.



Part Affinity Fields

Para lograr esto, los autores presentan una arquitectura de CNN especialmente diseñada para la detección de partes del cuerpo humano y una técnica de campos de afinidad para vincular las diferentes partes detectadas juntas para formar una pose completa. La arquitectura de CNN es basada en una red residual y utiliza varios módulos de escalabilidad para adaptarse a diferentes tamaños de personas en la imagen. La técnica de campos de afinidad utiliza un modelo de probabilidad para asociar las diferentes partes detectadas y mejorar la precisión de la estimación de la pose.

El sistema es evaluado en el dataset COCO (Common Objects in Context) y se compara con otros métodos, obteniendo una mejora significativa en términos de precisión y velocidad. Los autores demuestran que el sistema es capaz de detectar y seguir varias personas en una sola imagen con alta precisión y velocidad, a una tasa de hasta 25 frames por segundo en una sola GPU. Además, el sistema también es capaz de manejar situaciones de superposición de personas y poses extremadamente variadas, lo que lo hace adecuado para una variedad de aplicaciones.

En resumen, es una contribución importante en el campo del reconocimiento de poses de personas en tiempo real. El enfoque propuesto combina las ventajas de las CNNs y los campos de afinidad de partes para mejorar la precisión y velocidad del sistema. Este sistema tiene aplicaciones en diversas áreas, como el seguimiento de personas en entornos de vigilancia, el análisis de comportamientos en entornos sociales y el seguimiento de personas en entornos de robótica móvil.

4.2. Mobile Robot Navigation based on Deep Reinforcement Learning

El trabajo "Mobile Robot Navigation based on Deep Reinforcement Learning" (Ruan et al., 2019) propone un sistema de navegación autónoma para robots móviles basado en aprendizaje por refuerzo y deep neural networks (DNN). El sistema busca aprender una política de navegación, es decir, una serie de acciones para realizar en un entorno dado, mediante el uso de un algoritmo de aprendizaje por refuerzo.

En particular, los autores utilizan el algoritmo de Proximal Policy Optimization (PPO) para aprender la política de navegación. Este algoritmo es una variante del algoritmo de aprendizaje por refuerzo conocido como actor-critic y se utiliza para mejorar la estabilidad y eficiencia del aprendizaje en comparación con otros algoritmos de aprendizaje por refuerzo.

La información sensorial del robot, como las lecturas de sensores de movimiento, es procesada por una DNN para generar una acción de navegación para el robot. La DNN es entrenada mediante un proceso de entrenamiento en ambiente simulado y luego transferido al robot físico para su implementación en un ambiente real.

Los autores evalúan el sistema propuesto en un entorno simulado y un entorno real, comparando su rendimiento con un algoritmo de navegación basado en reglas. Los resultados muestran que el sistema propuesto basado en aprendizaje por refuerzo y DNN logra un rendimiento similar o mejor en comparación con el enfoque basado en reglas, especialmente en situaciones de incertidumbre y cambios en el ambiente. También se observa que el sistema es capaz de adaptarse a cambios en el entorno y aprender nuevas políticas de navegación de manera autónoma.

El trabajo "Mobile Robot Navigation based on Deep Reinforcement Learning" presenta un enfoque prometedor para la navegación autónoma de robots móviles que combina el poder de aprendizaje por refuerzo y las redes neuronales profundas. El sistema propuesto es capaz de aprender políticas de navegación eficientes y adaptarse a cambios en el entorno de manera autónoma, lo que lo hace adecuado para una variedad de aplicaciones en robótica móvil.

4.3. Exploración y reconocimiento de objetos basados en imágenes

El artículo "Aplicación de Deep Learning en robótica móvil para exploración y reconocimiento de objetos basados en imágenes" recoge el trabajo de unos investigadores de la Universidad de los Andes, en Bogotá, Colombia. El objetivo del trabajo era el de utilizar Deep Learning para reconocer grandes estructuras físicas del entorno (escaleras, pasillos, ascensores...) y que un robot móvil pudiera desplazarse de manera autónoma y segura por todo el espacio gracias a ese conocimiento de este.

A partir de la bio-inspiración del cerebro humano, las redes neuronales permiten que las máquinas puedan aprender sin necesidad de tener que programar todas las posibles situaciones que puedan encontrar, aspecto tedioso en una solución en el mundo real, ya que hay demasiadas variables que una función no puede calcular de forma completa o se tardaría mucho en ejecutar a través de un ordenador.

Deep Learning

La estructura pensada para el modelo de aprendizaje está separada en dos instancias, una para el reconocimiento de imágenes y la otra para el de navegación. Esto implica una etapa de entrenamiento que las dos redes reutilizan (con datos adecuados para cada reconocimiento), con una buena cantidad de datos de calidad.

Cada red tiene una configuración diferente. Una de las redes está enfocada en el proceso de aprendizaje de las diferentes estructuras físicas en un espacio interior, como escaleras, pasillos y ascensores. Esta red está compuesta por dos capas de neuronas que convolucionan la imagen, es decir, descomponen la imagen aplicando filtros para resaltar bordes de los objetos, y capas de *subsampling*, las cuales reducen la resolución de la imagen para reducir la carga de procesamiento y cinco capas de neuronas totalmente conectadas las cuales realizan la clasificación. La otra red aprende a predecir la dirección y la velocidad para moverse utilizando los datos de proximidad del sensor láser, es por esto que no tiene capas de convolución ni *subsampling*, ya que estos procesos son para tratamiento de imágenes.

BackPropagation

Para el proceso de entrenamiento y aprendizaje de la red neuronal se usa el algoritmo de BackPropagation, el cual a partir del resultado obtenido de una imagen o valor de entrada se genera un error con respecto al valor esperado y de esta manera, se retrocede por la red ajustando todos los pesos con relación a este error global.

Arquitectura del sistema

Como se ha comentado anteriormente, se plantea un sistema compuesto por las dos redes neuronales, el cual sea capaz de entrenar las redes en el modo Training. En tiempo real se obtiene la información de los sensores (cámara y láser), para poder reportar la información obtenida y generar acciones al robot (girar izquierda, girar derecha o mover al frente).

La red de reconocimiento de imágenes cuenta con 5 capas de 100 neuronas conectadas entre sí y una capa de salida de 5 neuronas relacionadas cada una con escaleras, pasillos anchos, pasillos estrechos, ascensores y un objeto de control, es decir, imágenes que no son las estructuras a reconocer. Esta configuración parece deberse a la gran cantidad de datos de píxeles a procesar, aunque requiere mayor poder de cómputo, la clasificación es más acertada y confiable. Para la red neuronal de navegación se cuenta con 3 capas de 25 neuronas conectadas entre sí y una capa de salida de 3 neuronas relacionadas con las posibles acciones de movimiento del robot.

Implementación y resultados

Según lo comentado en el artículo, se definieron 6 configuraciones independientes para las redes neuronales con entrenamientos diferentes en la navegación, en algunas impulsando la curiosidad del robot y otras manteniendo la seguridad. Luego de probar estas 6 redes neuronales en simulación y en el ambiente experimental, se obtuvo una red funcional, que se comportó muy bien en simulación, presentando 5 colisiones en todo el recorrido, navegando en el 60% aproximadamente del espacio.

Sin embargo, con el robot experimental solo se pudo recorrer un 40% del lugar, con 5 colisiones y muchas indecisiones al moverse. Esto se debió a que los datos obtenidos por el sensor láser de proximidad generan errores considerables comparados con lo que se simuló.

Con respecto al reconocimiento de estructuras (escaleras, pasillos anchos, pasillos estrechos y ascensores) a partir de imágenes no se presentaron muchos problemas.

Resultados del trabajo

Como conclusiones del artículo se identificó una diferencia importante entre la información obtenida por el sensor láser simulado (usado en entrenamiento) y el sensor láser real (usado en experimentación) que afectó la tasa de reconocimiento en la red neuronal de navegación. Indiferentemente los resultados parecieron satisfactorios y en definitiva se logró el objetivo de que un robot móvil se desarrollara de manera autónoma y segura en un entorno.

5. Experimentación de Deep Reinforcement Learning con Turtlebots

Para complementar lo visto en el apartado 4.2, se ha hecho una búsqueda de algún proyecto que implementase un sistema de aprendizaje por refuerzo profundo mediante el uso de turtlebots. Se ha encontrado un repositorio en GitHub en el que se ha creado un entorno de simulación en Gazebo para testear el funcionamiento del Deep Reinforcement Learning a la hora de realizar una tarea de navegación con robots móviles de tipo Turtlebot. El repositorio en cuestión se encuentra en el siguiente enlace:

<https://github.com/erdenbatuhan/autonomous-turtlebot/tree/master/results/old>

Este proyecto modela el problema a partir de un vector que almacena tuplas de distancia en cada instante de tiempo. Cada una de las tuplas contienen tanto el valor de la distancia entre el robot y el objetivo como el valor de la distancia entre el robot y una posible colisión. El primer elemento de las tuplas es un número, pero el segundo es otro vector de tres elementos que representa el valor medio de los píxeles de la imagen que proporciona la

cámara estéreo de profundidad. Son 3 columnas porque se ha dividido la imagen equitativamente en tres partes izquierda, centro y derecha.

Una vez se ha modelado el input que recibe el robot, los desarrolladores del experimento pasan a planificar tanto la arquitectura de la red neuronal como el algoritmo que va a usar dicha red para su aprendizaje. En primer lugar, se han escogido 4 redes neuronales, de las cuales 2 de ellas se van a usar para tomar acción a partir del primer elemento de las tuplas y las otras 2 para gestionar cómo actuar a partir del segundo elemento. Esto quiere decir que se divide el problema en 2, una parte que únicamente busca llegar al objetivo requerido en el mapa y otra que trata de que el robot no colisione. Para coordinar estos dos paradigmas lo que se hace es que actúen siempre las 2 redes que tratan de llegar al destino excepto cuando la distancia a un objeto sea muy pequeña y entonces intervendrán las otras dos redes que aportan seguridad al robot. La arquitectura de dichas redes será la siguiente:

- Redes que acortan la distancia al objetivo: 2 capas ocultas de 100 neuronas y una capa de salida de 3 neuronas (ir a la izquierda, a la derecha o avanzar). Un optimizador de tipo "Adam" y una tasa de aprendizaje con un valor de 0,01.
- Redes que buscan la seguridad: 3 capas ocultas de 100 neuronas y una capa de salida de 3 neuronas (ir a la izquierda, a la derecha o avanzar). Un optimizador de tipo "Adam" y una tasa de aprendizaje con un valor de 0,01.

Por otro lado, el algoritmo que utilizan estas redes neuronales para dar lugar al aprendizaje profundo por refuerzo es el "Deep Double Q-Learning", el cual es el motivo por el que se usan 2 redes neuronales para gestionar cada una de las dos acciones. Esto es así debido a que este algoritmo se basa en otro más primitivo que se usa en aprendizaje por refuerzo conocido como "Q-Learning" y que guarda en una tabla el valor que tiene tomar una acción en una determinada situación, sin embargo, esto a veces ocasiona un problema de sobreentrenamiento que se soluciona mediante la adición de una segunda red neuronal.

Los resultados que se obtienen en distintos entornos con distribuciones de obstáculos distintas presentan una muy notable mejoría respecto de agentes que actúan de forma aleatoria. Además, presentan una guía para poder probar su proyecto.

6. Referencias

Cao, Z., Simon, T., Wei, S.E., Sheikh, Y., & Sun, J. (2017). Real-time multi-person 2D pose estimation using part affinity fields. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1302-1310).

D. Liu, Z. Zhao, X. Wang, Y. Hu, L. Zhang and T. Huang, "Improving 3D Human Pose Estimation Via 3D Part Affinity Fields," 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2019, pp. 1004-1013, doi: 10.1109/WACV.2019.00112.

X. Ruan, D. Ren, X. Zhu and J. Huang, "Mobile Robot Navigation based on Deep Reinforcement Learning," 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 6174-6178, doi: 10.1109/CCDC.2019.8832393.

K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," in Tsinghua Science and Technology, vol. 26, no. 5, pp. 674-691, Oct. 2021, doi: 10.26599/TST.2021.9010012.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2017). Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6), 1452-1464.

S. Contreras and F. De la Rosa, "Aplicación de deep learning en robótica móvil para exploración y reconocimiento de objetos basados en imágenes," 2016 IEEE 11th Colombian Computing Conference (CCC), Popayan, Colombia, 2016, pp. 1-8, doi: 10.1109/ColumbianCC.2016.7750800.